Statistical Methods for Machine Learning

Sunspots, Seeds, and Stars

Exam Assignment

Christian Igel and Sami Brandt
Department of Computer Science
University of Copenhagen

This is the final exam assignment on the course *Statistical Methods for Machine Learning*, block 3 2012, at the University of Copenhagen. It is based on the full course curriculum as stated in the lectures schedule in the Absalon system. The assignment is centered around real world pattern recognition tasks.

This assignment must be made and submitted *individually*. However, it is acceptable (and we encourage) that you discuss the solution of the assignment with your fellow students. It is also acceptable (and we encourage) to use bits and pieces of your solutions and the hand-out code from the previous assignments.

Your solution to this assignment will be graded using the 7-point scale and will be the final grade for the course. To obtain the best grade of 12, you must fulfill all the course learning objectives (see below) at an excellent level. In terms of the questions in this assignment, this means that you have to answer all questions with non or only a few mistakes or parts missing. To obtain the passing grade of 02 you need to fulfill the learning objectives at a minimum level, which means you have to make a serious attempt at solving the central questions in the assignment (but not necessarily all) with some mistakes allowed.

The deadline for this assignment is **April 4, 2013**. You must submit your solution electronically via the Absalon home page. Go to the assignments list and choose this assignment and upload your solution prior to the deadline.

Solution format

The deliverables for each question are listed at the end of each question. The deliverable "description of software used" means that you should hand in the source code you have written to solve the problem. If you have used a software library to solve the problem, this library should be described and reasons for the particular choice should be given.

Thus, a solution should contain:

- A PDF document showing your results and giving detailed answers to the questions. If relevant, this may include graphs and tables with comments (**max. 10 page of text including figures and tables**). Use meaningful labels, captions, and legends. Do **not** include your source code in this PDF file. You will be graded mainly on the basis of this report.
- Your solution code (Matlab / R / Python scripts or C / C++ code) with comments about the major steps involved in each question. The code must be submitted in its original format (e.g., in `.m` or `.R` file format – not as PDF files). Use meaningful names for files, constants, variables, functions and procedures etc. Add comments to the code to make it more readable.
- Your code should be structured such that there is one main file that we can run to reproduce all the results presented in your report. This main file can, if you like, call other files with functions / classes.
- Your code should also include a README text file describing how to compile (if relevant) and run your program, as well as a list of all relevant libraries needed for compiling or using your code. If we cannot make your code run we will consider your submission incomplete.

Learning objectives

The grade will be based on a judgement of how well you fulfill the following learning objectives:

1. Recognize and describe possible applications of machine learning for pattern recognition and data mining.
2. Explain, contrast and apply basic Bayesian probability theory for modeling stochastic data, including both parametric and non-parametric representations.
3. Explain and contrast the concept of supervised and unsupervised learning.
4. Explain the concepts of classification and clustering.
5. Identify, explain and handle the common pitfalls of machine learning.
6. Describe and apply linear techniques for classification.
7. Implement selected machine learning techniques.
8. Use software libraries for solving machine learning problems.

9. Visualize and evaluate results obtained with a machine learning method.
10. Compare, appraise and select methods of machine learning for solving specific problems of pattern recognition and data mining.

Case 1: Sunspot Prediction

Background

The goal is to predict the average number of sunspots. We consider data based on the yearly sunspot number data provided by the Sunspot Index Data Center (SIDC), see `http://sidc.oma.be`. On `http://www.icsu-fags.org/ps11sidc.htm` we find the following short introduction to the sunspot data:

> Sunspots are extended regions on the Sun with a strong magnetic field. They have a lower temperature (3500-4500 K) than the surrounding photosphere (5800 K). The sunspots radiate less energy than the undisturbed photosphere of the Sun and are therefore visible as dark spots on the surface of the Sun. Sunspots are observed with some regularity since 1700 and on a strict daily basis since 1849; the relative [...] number (defined as ten times the number of groups + the number of spots) shows an 11 year cycle detected by Schwabe in 1843. The sunspot number reflects the magnetic activity of the Sun, which has a large impact to the magnetosphere of the Earth and is responsible for e.g. magnetic storms, polar lights.

In this assignment, the task is to predict the average number of sunspots in a year t based on the average numbers in the years $t-1$, $t-2$, $t-4$, $t-8$, and $t-16$.

The training data is in the file `sunspotsTrainStatML.dt` and the test data on `sunspotsTestStatML.dt`. Each row correspond to five input features (predictor variables) and, in the last column, the desired target (response variable). The targets in the training data are the years 1716–1915. The targets in the test data are the years 1916–2011.

The Exercises

Question 1 (linear regression)**.** The goal of our modeling is to find a mapping $f : \mathbb{R}^4 \to \mathbb{R}$ for predicting the number of sunspots based on previous observations.

Build an affine linear model of the data using linear regression and the training data in `sunspotsTrainStatML.dt` only. Report the six parameters of the model.

Determine the training error by computing the mean-squared-error of the model over the complete *training* data set. Compute the mean-squared-error on the test data set `sunspotsTestStatML.dt`. Comment very briefly on the result.

Deliverables: description of software used; parameters of the regression model; mean-squared error on the training and test data set; short discussion of results

Question 2 (non-linear regression)**.** Now apply a non-linear regression method covered in the lecture to the data. Describe the measures you have taken to ensure good generalization. Use the same split in training and test data, and use the mean-squared error to evaluate your model.

Deliverables: description of software used; mean-squared error on the training and test data set; describe what you did to get good generalization performance; discussion of results

Question 3 (visualization)**.** Plot the sunspots time series from year 1916 to 2011 (years on the x-axis, average number of sunspots on the y-axis). Add the predictions of your linear and non-linear model (i.e., the predictions you used to compute the test errors) to the plot to visualize the quality of your model. Comment briefly on the result.

Deliverables: visualization of the test data (number of sunspots over time) and the corresponding outputs of the linear and the non-linear model; short discussion of the plot

Case 2: Surveying the Sky

Background

Astrophysics and cosmology are rich with data. The advent of wide-area digital cameras on large aperture telescopes has led to ever more ambitious surveys of the sky. The data volume of an entire survey of a decade ago can now be acquired in a single night and real-time analysis is often desired. The magnitude of these surveys makes manual examination impossible – and this is where machine learning comes in.

We consider data from *The Sloan Digital Sky Survey*, see `http://www.sdss.org`. The taks is to distinguish between stars and quasars ("quasi-stellar radio sources").

Quasars are the brightest objects in the universe. Because of their huge distance, they look like single bright points in photo images and are therefore hard to distinguish from stars in such images. However, they can be identified by looking at the spectrum of the light they emit. The data stem from a telescope that images the sky in five different wavelengths of light simultaneously, see Table 1.

Name	Color	Wavelength
u’	ultraviolet	3540 Å
g’	blue/green	4760 Å
r’	red	6280 Å
i’	infrared	7690 Å
z’	infrared	9250 Å

Table 1: SDSS filters leading to the features in the quasar classification task.

Training and test data are stored in the files `quasarsStarsStatMLTrain.dt` and `quasarsStarsStatMLTest.dt`, respectively. Each row correspond to five input features (the five filter responses) and, in the last column, the desired binary target. A 0 encodes a star, a 1 encodes a quasar.

The Exercises

Question 4 (binary classification using support vector machines)**.** The task is to perform binary classification using support vector machines (SVMs). For this exercise, use standard C-SVMs as introduced in the lecture. Employ radial Gaussian kernels of the form

$$k(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{z}\|^2) \ .$$

Here $\gamma > 0$ is a bandwidth parameter that has to be chosen in the model selection process. Note that instead of γ often the parameter $\sigma = \sqrt{1/(2\gamma)}$ is considered.

Jaakkola’s heuristic provides a reasonable initial guess for the bandwidth parameter σ or γ of a Gaussian kernel [Jaakkola et al., 1999]. To estimate a good value for σ, consider all pairs consisting of a training input vector from the positive class and a training input vector from the negative class. Compute the difference in input space between all pairs. The median of these distances can be used as a measure of scale and therefore as a guess for σ. More formally, compute

$$G = \{\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \,|\, (\boldsymbol{x}_i, y_i), (\boldsymbol{x}_j, y_j) \in S \wedge y_i \neq y_j\}$$

based on your training data S. Then set σ_{Jaakkola} equal to the median of the values in G:

$$\sigma_{\text{Jaakkola}} = \text{median}(G)$$

Compute the bandwidth parameter γ_{Jaakkola} from σ_{Jaakkola} using the identity given above.

Use grid-search to determine appropriate SVM hyperparameters γ and C. Look at all combinations of

$$C \in \{b^{-1}, 1, b, b^2, b^3\}$$

and

$$\gamma \in \left\{\gamma_{\text{Jaakkola}} \cdot b^i \,|\, i \in \{-3,-2,-1,0,1,2,3\}\right\} \ ,$$

where the base b can be chosen to be either 2, the base e of the natural logarithm (Euler's number), or 10. Feel free to vary this grid. For each pair, estimate the performance of the SVM using 5-fold cross validation (see section 1.3 in the textbook by Bishop [2006]). Pick the hyperparameter pair with the lowest average 0-1 loss (classification error) and use it for training an SVM with the complete training dataset. Only use the data from `quasarsStarsStatMLTrain.dt` in the model selection and training process.

Report the values for C and γ you found in the models selection process. Compute the classification accuracy based on the 0-1 loss on the training data as well as on the test data.

Deliverables: description of software used; a short description of how you proceeded; initial γ or σ value suggested by Jaakkola's heuristic; optimal C and γ found by grid search; classification accuracy on training and test data

Question 5 (overfitting)**.** John Langford, who is "Doctor of Learning at Microsoft Research", maintains a very interesting blog (web log). Read the very true blog entry: "Clever methods of overfitting," `http://hunch.net/?p=22`, 2005. Discuss if and how the different types of overfitting can occur when applying machine learning techniques to the sunspot regression task and the quasar classification task. Ignore the last type of overfitting. You need not discuss issues related to reviewing of scientific papers (still, it is good to keep them in mind).

Deliverables: short discussion addressing the first 10 "methods of overfitting" listed in the blog entry

Case 3: Wheat Seeds

Background

This classification task addresses the problem of identifying a variant of wheat based on properties of its seed. The seed from which the wheat plant grows is called wheat kernel or wheat beery. We consider data available from the well-known UCI benchmark repository (Frank and Asuncion, 2010, `http://archive.ics.uci.edu/ml/datasets/seeds`). Charytanowicz et al. [2010] measured seven geometrical properties of kernels belonging to three different types of wheat, namely *Kama*, *Rosa*, and *Canadian*. The seven features are listed in see Table 2.

The files `seedsTrain.dt` and `seedsTest.dt` contain the training and test data, respectively. The inputs are the seven geometric features and the integer in the last column encodes the wheat variant.

Table 2: Brief description of the input attributes for wheat classification, see the article by Charytanowicz et al. [2010] for details.

feature index	description
0	area A
1	perimeter P
2	compactness $C = 4\pi A/P^2$
3	length of kernel
4	width of kernel
5	asymmetry coefficient
6	length of kernel groove

The Exercises

Question 6 (principal component analysis)**.** Perform a principal component analysis of the training data in `seedsTrain.dt`. Plot the eigenspectrum (see Figure 12.4 by Bishop [2006] for an example). Visualize the data by a scatter plot of the data projected on the first two principal components. Use different colors for the different classes in the plot (see Figure 12.8 by Bishop [2006] for an example, which, however, lacks proper axes labels).

Deliverables: description of software used; plot of the eigenspectrum; scatter plot of the data projected on the first two principal components with different colors indicating the 7 different classes

Question 7 (clustering)**.** Perform 3-means clustering of the training data in `seedsTrain.dt` and report the cluster centers. *After that*, project the cluster centers to the first two principal components of the training data. Then visualize the clusters by adding the cluster centers to the plot from the previous exercise. Briefly discuss the results: Did you get meaningful clusters?

Deliverables: description of software used; cluster centers; one plot with cluster centers and data points; short discussion of results

Question 8 (multi-class classification)**.** Use a linear and a non-linear classification method (picking from the methods presented in the course) for classifying the 7 image classes, for example k-nearest neighbor and linear discriminant analysis (LDA). Only use the training data in `seedsTrain.dt` in the model building process. After you trained a model, use the test data in `seedsTest.dt` to evaluate it. Report the classification error on both training and test set.

Deliverables: description of software used; arguments for your choice of classification methods; a short description of how you proceeded and what training and test results you achieved

References

C. Bishop. *Pattern Recognition and Machine Learning.* Springer, 1 edition, 2006.

M. Charytanowicz, J. Niewczas, P. Kulczycki, P. A. Kowalski, S. Łukasik, and S. Żak. Complete gradient clustering algorithm for features analysis of x-ray images. In E. Piętka and J. Kawa, editors, *Information Technologies in Biomedicine*, volume 69 of *Advances in Intelligent and Soft Computing*, pages 15–24. Springer, 2010.

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

T. Jaakkola, M. Diekhaus, and D. Haussler. Using the Fisher kernel method to detect remote protein homologies. In T. Lengauer, R. Schneider, P. Bork, D. Brutlad, J. Glasgow, H.-W. Mewes, and R. Zimmer, editors, *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158. AAAI Press, 1999.